UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/817,288	03/27/2001	Yoshitake Shinkai	826.1718	7742

21171 7590 08/03/2007

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
LESNIEWSKI, VICTOR D	
ART UNIT	**PAPER NUMBER**
2152	
MAIL DATE	**DELIVERY MODE**
08/03/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/817,288
Filing Date: March 27, 2001
Appellant(s): SHINKAI ET AL.

Reginald D. Lucas, Reg. No. 46883
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 5/1/2007 appealing from the Office action mailed 7/12/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct. A request for reconsideration was filed after final on 11/13/2006, but the request did not present any amendments to the claims. The request was addressed in the advisory action dated 12/8/2006.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,964,886	SLAUGHTER ET AL.	10-1999
5,634,122	LOUCKS ET AL.	5-1997
5,515,537	TAVARES ET AL.	5-1996

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-3, 8, 10-12, 14, 23, 25, 27, and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Slaughter et al. (U.S. Patent Number 5,964,886) in view of Loucks et al. (U.S. Patent Number 5,634,122), hereinafter referred to as Loucks.

Slaughter disclosed a virtual disk system that provides consistent data to a plurality of nodes in a cluster. In an analogous art, Loucks disclosed a distributed file system that uses a local token manager to resolve token conflicts before granting tokens.

Concerning the independent claims, Slaughter did not explicitly state a token manager that gives access permission for a file when no other node has update permission. Although Slaughter's system does set forth different types of access permissions, the system is mainly focused on accessing storage devices found to be available or in an active state. However, Loucks remedies the token manager issue as his system (also focused on accessing a file on a storage device) utilizes a manager to grant or deny tokens to requesting network nodes. Similarly, since Slaughter does not deal explicitly with tokens, Slaughter also does not state notifying the requesting node of another node that has a token. However, Slaughter does operate similarly in that his system maintains a list of active devices and when needed informs the requesting node of an alternate node through which the requested file can be accessed. Thus, it would be a simple extension of Slaughter's system to use a token manager, such as the one presented by Loucks, in conjunction with or instead of the active device list and then base the system on a required token. Said another way, Slaughter's system accesses file storage devices based on availability. Slaughter explicitly discusses availability in terms of an active network connection or the failure of a network connection. However, another type of availability is whether or not the device is available based on which node has a required token. Here, Loucks' token system can be utilized with Slaughter to determine a certain availability while Slaughter's system would still maintain its operating logic in accessing the storage devices. For more detail, see the specific line citations below. Thus, it would have been obvious to one of ordinary skill in

the art at the time of the applicant's invention to modify the system of Slaughter by adding the ability to utilize a token manager as claimed (for such purposes as giving access permission for a file when no other node has update permission and notifying the requesting node of another node that has a token) as provided by Loucks. Here the combination satisfies the need for a virtual disk system that ensures that each node has consistent permission data. See Slaughter, column 2, lines 25-29. This rationale also applies to those dependent claims utilizing the same combination.

Thereby, the combination of Slaughter and Loucks discloses:

- <Claims 1 and 10>

  (presented together since they disclose similar limitations)

  A file replication system having a plurality of nodes connected to a network, files being distributed to the nodes, wherein a first node of the nodes comprises: a first token managing portion giving access permission for a file within the first node when no other node has update permission and otherwise issuing a notification of a permitted node that has update permission for the file in response to an access request in first node (Slaughter, column 9, lines 24-27 and 36-47 and Loucks, figure 8b), and an IO request intercepting portion accepting an access to the file, the access taking place in the first node when said IO request intercepting portion is capable of acquiring the access permission, asking said first token managing portion to acquire the access permission against the access request, and asking the permitted node that has update permission for the file to access to the file when said first token managing portion is not capable of acquiring the access permission (Slaughter, column 9, lines 27-35; column 10, lines 16-

28; and column 8, lines 13-26), and a second node comprises a second token managing portion notifying a requesting node that requests the access permission for the file of the permitted node that has the update permission for the file as a response message (Slaughter, column 8, lines 33-41 and Loucks, column 6, lines 36-49).

- <Claims 2 and 11>

  A node, connected to at least one other node through a network, every node having a copy of files synchronized with files of other nodes for high availability and high performance to provide a replicated file system, comprising: a token managing portion managing an access request for a file (Loucks, column 6, lines 36-49); and an IO request intercepting portion asking said token managing portion to acquire an access permission for the file against an access request to the file in said node, said token managing portion giving access permission when no other node has update permission for the file and said token managing portion notifying said IO request intercepting portion of another node that has the update permission when the other node has the update permission for the file, in response to the access request of said IO request intercepting portion, said IO request interception portion accessing the file in said node when the IO request intercepting portion is capable of acquiring the access permission and said IO request intercepting portion asking the other node that has the update permission to access the file instead of accessing the file in said node when said IO request intercepting portion is not capable of acquiring the access permission (Slaughter, column 9, lines 24-47; column 10, lines 16-28; and column 8, lines 13-26 and Loucks, figure 8b).

- <Claim 3>

  The node according to claim 2, further comprising: a system structure managing portion performing a restoration process of data of a file of the node when it is newly joined to a system (Slaughter, column 8, lines 42-49), wherein while said system structure managing portion is restoring the file, when an access request for the file takes place in the node, said IO request intercepting portion asks another node that shares the file to access the file (Slaughter, column 10, lines 30-43).

- <Claims 8 and 12>

  A node, connected to at least one other node through a network, every node having a copy of files synchronized with files of other nodes for high availability and high performance to provide a replicated file system, comprising: a token managing portion asking another node to acquire an access permission for a file against an access request for the file in said node (Slaughter, column 9, lines 27-35 and column 10, lines 16-28); and an IO request intercepting portion accepting an access request for a file in said node, asking said token managing portion to acquire the access permission for the file against the access request to the file in said node, said token managing portion giving access permission when no other node has update permission for the file and otherwise notifying said IO request intercepting portion of another node that has the update permission for the file, said IO request intercepting portion accessing the file in said node when the IO request intercepting portion is capable of acquiring the access permission and asking the other node that has the update permission for the file to access the file according to the access request instead of accessing the file in said node when said token managing

portion is not capable of acquiring the access permission for the file (Slaughter, column 9, lines 24-47; column 10, lines 16-28; and column 8, lines 13-26 and Loucks, column 6, lines 23-49 and figure 8b).

- <Claim 14>

  A file replication control method for a system having a plurality of nodes connected to a network, files being distributed to the nodes, comprising: causing an access requesting node to access a file of the access requesting node itself when the access requesting node has the latest data of the file and has or is able to obtain access permission from another node having update permission for the file; and asking the other node to access the file when the other node has the update permission for the file which is given to only one node at a time (Slaughter, column 9, lines 24-47 and column 10, lines 16-28 and Loucks, column 6, lines 36-49 and figure 8b).

- <Claim 23>

  The file replication control method according to claim 14, further comprising: restoring data of a file of a newly joined node (Slaughter, column 8, lines 42-49); and operating a user program before data of the file is completely restored (Slaughter, column 10, lines 30-43).

- <Claim 25>

  The file replication control method according to claim 23, wherein the node asks another node that shares the file to perform a process for an access request for the file when the access request takes place in the node itself before data is completely restored (Slaughter, column 10, lines 30-43).

- <Claim 27>

  A file replication method for a system having a plurality of nodes connected to a network, files being distributed to the nodes, comprising: causing a first node to request a token for accessing a file (Loucks, column 6, lines 36-49); causing the first node to access the file when the first node has the latest data of the file and is able to obtain the token for accessing the file from another node having update permission for the file which is given to only one node at a time (Slaughter, column 9, lines 24-27 and 36-47 and Loucks, figure 8b); notifying the first node of a second node that has the token when the first node is not capable of acquiring the token (Slaugher, column 8, lines 33-41 and Loucks, column 6, lines 36-49); and causing the first node to ask the second node to access the file when the first node is notified that the first node is not capable of acquiring the token (Slaughter, column 9, lines 27-35 and column 10, lines 16-28).

- <Claim 28>

  A computer-readable portable storage medium, when being used by a computer that composes a node connected to other nodes through a network in a file replication system, on which is recorded a program for causing the computer to execute a process, said process comprising: when the node accesses a file and a node itself has the latest data of the file and has or is able to obtain access permission from another node having update permission for the file, causing the node itself to access the shared file of the node itself (Slaughter, column 9, lines 24-27 and 36-47 and Loucks, figure 8b); and when another node has the update permission for the file which is given to only one node at a time,

causing the node itself to ask the other node to access the file (Slaughter, column 9, lines 27-35 and column 10, lines 16-28 and Loucks, column 6, lines 36-49).

Since the combination of Slaugher and Loucks discloses all of the above limitations, claims 1-3, 8, 10-12, 14, 23, 25, 27, and 28 are rejected.

Claims 4-7, 16-22, 24, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Slaughter in view of Loucks, as applied above, further in view of Tavares et al. (U.S. Patent Number 5,515,537), hereinafter referred to as Tavares.

The combination of Slaughter and Loucks disclosed a virtual disk system that provides consistent data to a plurality of nodes in a cluster and uses a token manager to resolve token conflicts before granting tokens. In an analogous art, Tavares disclosed a method of sharing data between processors on the same network by using a real-time distributed locking system. Both systems focus on data sharing and methods to allow users access to certain data.

Concerning claims 4, 16, 17, and other related dependents such as claims 21, 22, and 24, although the combination of Slaughter and Loucks states updating content in each node of the virtual disk system, it did not explicitly state propagating updated content of the file to other nodes as claimed or the update's dependent relationship between nodes as claimed. However, Tavares does state a system wherein update data is propagated throughout the system as claimed. See figure 7. Furthermore, Tavares sends updated content and also passes a token from one node to the next in an ordered fashion based on the physical architecture of the system or on the processing of queues. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to modify the combination of Slaughter and Loucks by adding the

ability to propagate updated content of the shared file to other nodes and utilize an update's dependent relationship between nodes as provided by Tavares. Here the combination satisfies the need for a file replication system that ensures that each node has consistent permission data. See Slaughter, column 2, lines 25-29. This rationale also applies to those dependent claims utilizing the same combination.

Thereby, the combination of Slaughter, Loucks, and Tavares discloses:

- <Claim 4>

  The node according to claim 2, further comprising: a changed data notifying portion propagating an updated content of the file to other node along with information that represents a dependent relationship with another update (Tavares, column 6, lines 55-59); and a received data processing portion reflecting the updated content to the shared file while assuring an order of the update based on the dependency relationship (Tavares, column 8, lines 12-15).

- <Claim 5>

  The node according to claim 4, further comprising: a system state information portion storing information about propagation mode of an updated content for each of at least one file, wherein said changed data notifying portion propagates the update content based on information queued in said system information portion (Tavares, column 9, line 60 through column 10, line 20).

- <Claim 6>

  The node according to claim 5, wherein the propagation mode is one of a synchronous mode in which it is assured that the updated content is propagated to all the nodes that

share the file, a semi-synchronous mode in which it is assured that the updated content is propagated to the majority of nodes that share the file, and an asynchronous mode in which it is not acknowledged that the updated content is propagated to the nodes that share the file (Tavares, column 9, lines 60-66).

- <Claim 7>

  The node according to claim 4, wherein said system state information storing portion keeps information about each node that shares at least one file for each file (Tavares, column 10, lines 6-9).

- <Claim 16>

  The file replication control method according to claim 14, wherein the other node that has the update permission releases the update permission after an update that has a dependent relationship with the update performed at the other node has been propagated to all the nodes (Tavares, column 7, lines 37-39).

- <Claim 17>

  The file replication control method according to claim 14, wherein said method further comprises: the other node that has updated the file asynchronously propagating an updated content to the other nodes; and causing the other node that has updated the file to process an access request that takes place in the access requesting node while the updated content is being propagated (Tavares, column 10, lines 15-20).

- <Claim 18>

  The file replication control method according to claim 17, wherein the updated content is reflected in such a manner that order thereof is assured (Tavares, column 7, lines 2-16).

- <Claim 19>

  The file replication control method according to claim 18, wherein a dependency information that represents order of other updates to be propagated to the other node along with the updated content (Tavares, column 8, lines 12-15).

- <Claim 20>

  The file replication control method according to claim 19, wherein a node that has received the updated content to reflect the updated content on a file of the node itself after receiving a previous updated content based on the dependency information (Tavares, column 10, lines 15-20).

- <Claim 21>

  The file replication control method according to claim 14, wherein a propagation mode of an updated content is designated for each of at least one file (Tavares, column 9, lines 60-66).

- <Claim 22>

  The file replication control method according to claim 14, wherein a node to which an updated content is propagated is designated for each of at least one file (Tavares, column 7, lines 2-16).

- <Claim 24>

  The file replication control method according to claim 23, wherein restored data is transmitted in such a manner that order of update requests for the file is assured (Tavares, column 8, lines 12-15).

- <Claim 26>

  The file replication control method according to claim 14, wherein a node that has performed a systematic stop in which nodes that share a file are synchronously stopped to store a systematic stop state and the node synchronously resumes a process for the file without restoring data of the file (Tavares, column 9, lines 60-66).

Since the combination of Slaughter, Loucks, and Tavares discloses all of the above limitations, claims 4-7, 16-22, 24, and 26 are rejected.

(10) Response to Argument

In the brief, the appellant has argued:

- <Argument 1>

  The combination of Slaughter and Loucks does not disclose the features of independent claim 1 and like independent claim 10 because it does not disclose "an IO request intercepting portion accepting an access to the file, the access taking place in the first node when said IO request intercepting portion is capable of acquiring the access permission" as recited in claim 1.

In response to argument 1 (set forth on pages 6-7 of the brief under heading A3), the combination of Slaughter and Loucks does disclose an IO request intercepting portion accepting an access of the file in the first node as recited in claim 1. The previous line citation to Slaughter, column 9, lines 27-35, states that the netdisk driver routes the data access request to the primary node (i.e. the first node) when the primary node is active. This functionality is seen

to meet the limitation at hand. In support of the argument, the appellant has stated that "in Slaughter, file access does not actually occur in the netdisk driver 318A, as the net disk driver 318A forwards the data access request to either a primary node or a secondary node." This statement, however does not seem to relate directly to the claim language which requires that the access take place in the first node. Although the IO request intercepting portion is part of the first node, the claim does not require that the access take place within this portion, only in the first node as a whole. To this it is shown that Slaughter's "intercepting portion" (or the netdisk driver) is directly coupled to the primary node (i.e. it is part of the first node). Slaughter may use the netdisk driver to perform the access via the netdisk master in the primary node (thus, accepting access of the file in the first node) or may forward the request to a separate node not directly coupled via the cluster transport interface. See Slaughter, column 9, lines 36-61, column 13, lines 19-34, and figure 6, steps 618-624.

- <Argument 2>

  The combination of Slaughter and Loucks does not disclose the features of independent claim 1 and like independent claim 10 because it does not disclose "asking the permitted node that has update permission for the file to access the file" as recited in claim 1.

In response to argument 2 (set forth on page 7 of the brief under heading A3), the combination of Slaughter and Loucks does disclose asking the permitted node to access the file as recited in claim 1. The previous line citation to Slaughter, column 9, lines 27-35, states that Slaughter's system sends the data access request to a second node when the first node is not

active. The previous line citation to Slaughter, column 10, lines 16-28, further describes the processing of the data access request while the previous line citation to Slaughter, column 8, lines 13-26, shows the maintenance of a list of active nodes in the system. In terms of the argued limitation, Slaughter's primary node handles the data access request when active (meaning "access request taking place in the first node"). The node is active when it has membership information for the cluster (meaning the node is permitted in the cluster and is capable of accessing the data). When the primary node is not active (meaning "not capable of acquiring the access permission"), the data access request is routed to a secondary node which is active (meaning "asking the permitted node to access to the file"). This secondary node is considered "permitted" because it is active which means the node has membership information for the cluster and is capable of accessing the data.

Despite this, the appellant has stated that "Slaughter simply transmits the access request to another node, without regard to what entity has update permission for a file. The other node is not necessarily the node that has access permission. That is, the node to which Slaughter transmits the data access request may then forward the data access request to yet another node." First, it is noted that in attempting to illustrate this point in the appellant's example scenario, the netdisk driver 318A, the NM 320A, and the disk driver 326A are all portions coupled to the same primary node, not distributed across multiple nodes. Second, it is noted that Slaughter does not transmit the request to another node without regard to permissions. The point of Slaughter's system is to transmit the request to another node which is active when the first is not active, meaning that Slaughter's system is aware of permissions since it must pass the request to a node capable of accessing the data when the first node is not capable. A node is capable of accessing

the data in Slaughter's system when it has membership information which allows that node access to the data. Third, it is noted that even if Slaughter's system were to transmit the access request to a second node which then has to transmit it on to yet another node, as the appellant purports, this scenario would still meet the limitation at hand as the claim does not state that the request must be transmitted directly to the second node, the claim only states asking a permitted node to access the file.

- <Argument 3>

  The combination of Slaughter and Loucks does not disclose the features of independent claim 2 and like independent claims because it does not disclose "asking the other node that has the update permission to access the file" as recited in claim 2.

In response to argument 3 (set forth on page 8 of the brief under heading B2), the combination of Slaughter and Loucks does disclose asking the other node that has the update permission to access the file as recited in claim 2. This argument is substantially the same as argument 2 and thus it is maintained that the combination of Slaughter and Loucks teaches these features for the same reasons as given above. See the response to argument 2 above.

- <Argument 4>

  The combination of Slaughter and Loucks does not disclose the features of independent claim 8 and like independent claims because it does not disclose "said token managing

portion giving access permission when no other node has update permission for the file" as recited in claim 8.

In response to argument 4 (set forth on page 9 of the brief under heading B2), the combination of Slaughter and Loucks does disclose the token managing portion giving access permission when no other node has update permission for the file as recited in claim 8. The previous line citation to Loucks, column 6, lines 22-49, states that Loucks's token manager can issue tokens, maintain a list of tokens, revoke tokens, etc. In addition see Loucks, figure 8b, also previously cited. One purpose of Loucks's system is to resolve token conflicts before granting tokens, meaning the system will not give permission to one node when another node already has permission for the same file.

In support of the argument, the appellant has stated that Slaughter does not teach the limitation at hand. However, the rejection states that the teachings of Loucks were cited in support of the limitation. See the description of the combination of Slaughter and Loucks in the rejection above. The appellant has not referred to any of the lines cited to Loucks in the arguments and thus it is unclear as to why the appellant believes the combination of Slaughter and Loucks does not teach the claimed limitations. The appellant is reminded that one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

- <Argument 5>

  The combination of Slaughter and Loucks does not disclose the features of independent claim 14 and like independent claims because it does not disclose "causing an access requesting node to access a file of the access requesting node itself when the access requesting node has the latest data of the file and has or is able to obtain access permission from another node having update permission for the file" as recited in claim 14.

In response to argument 5 (set forth on pages 9-10 of the brief under heading C2), the combination of Slaughter and Loucks does disclose causing an access requesting node to access a file of the access requesting node itself as recited in claim 14. This argument is substantially the same as argument 1 and thus it is maintained that the combination of Slaughter and Loucks teaches these features for the same reasons as given above. See the response to argument 1 above.

For the above reasons, it is believed that the rejections should be sustained.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

Respectfully submitted,

Victor Lesniewski
Patent Examiner
Group Art Unit 2152

Dated: July 25, 2007

Conferees:

Lynne H. Browne
Appeal Practice Specialist, TQAS
Technology Center 2100

BUNJOB JAROENCHONWANIT
SUPERVISORY PATENT EXAMINER 7/31/7